



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/785,625	02/16/2001	Christine Michelle Barnes	343355600026	4951

7590 03/15/2004  
John V. Biernacki  
Jones Day, Reavis & Pogue  
North Point  
901 Lakeside Avenue  
Cleveland, OH 44114

EXAMINER

KANG, INSUN

ART UNIT PAPER NUMBER

2124

DATE MAILED: 03/15/2004

2

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No. 09/785,625	Applicant(s) BARNES ET AL.	
	Examiner Insun Kang	Art Unit 2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 16 February 2001.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-31 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-31 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 16 February 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. This action is in response to the application filed 2/16/2001.
2. Claims 1-31 are pending in the application.

### ***Specification***

3. The use of the trademark JAVABEANS and JAVA has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

### ***Claim Rejections - 35 USC § 112***

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 7, 15 and 17 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "substantially" in claim 7 is a relative term which renders the claim indefinite. The term "substantially" is not defined by the claim, the specification does

not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

Claim 15 recites the limitation "the public object state data" in line 11. There is insufficient antecedent basis for this limitation in the claim.

Per claim 17, the claim uses the term "capable." It is unclear whether the computer-readable file is actually being imported or it is simply "capable" of being imported without any actions. The former is used for interpretation.

### ***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-31 are rejected under 35 U.S.C. 102(b) as being anticipated by Johnson (XML JavaBeans series, Part 1-3, published 2-7/1999, JavaWorld).

Per claim 26:

Johnson discloses:

- generating a node tree whose nodes store the public and private object state data ("an XML document in a program can be represented as a tree of ... (DOM) nodes...flatten that tree of DOM tree and write it to a text file," pg 5-6 in Part 2; Simpler XML in Part 3; "call the method getAsDOM(), which builds a **DOM**

**document tree based on the properties of the JavaBean,**" pg 7 par 4-5 in Part 2; see pg 6 par 4-5 in Part1; pg 9 paragraph 4-5 in Part 3).

- processing the nodes of the node tree to generate nodes in an XML tree, wherein the nodes in the XML tree correspond to an XML tag structure ("convert a JavaBean to a tree, and then convert the tree to text, which is then written to a file... The XML corresponding to the DOM tree," pg 5-6 in Part 2; see Figure 3 in Part 2; pg 8 in Part 1)
- generating XML tags based upon the nodes in the XML tree, wherein the XML tags are structured so as to persist the public and private object state data ("XML as a persistence format for JavaBeans components," pg 2 in Part 2; "XMLBeanReader gives...the ability to take a "flat" representation of an object structure, that is, a JavaBean and its properties represented as a text stream that just happens to be XML," pg 3 par. 3 in Part 2; pg 5-6 in Part 2; see pg 6 par 4-5 in Part1; see Figure 4).

Per claim 27:

The rejection of claim 26 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the public and private object state data ("readXMLBean ... creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3)
- instantiating objects based upon the recovered public and private object state data ("identifies the first element of the document read, and calls

- instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)
- using the instantiated objects within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3; "controlling XMLBeans properties by integrating XMLBeans with the core java.beans package," pg 2 par 1 in Part 3).

Per claim 28:

The rejection of claim 27 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the design time object state data ("readXMLBean ... creates an XML parser, reads the input the input XML file," pg 12 par 3 in Part 3);
- instantiating objects based upon the recovered design time object state data ("identities the first element of the document read, and calls instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)
- using the instantiated objects within the object development environment such that the recovered design time object state data is used only within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3).

Per claim 29:

The rejection of claim 26 is incorporated, and further, Johnson teaches:

Art Unit: 2124

- parsing the XML tags to recover the public and private object state data  
(“readXMLBean ... creates an XML parser, reads the input the input XML file,” pg 12 par 3 in Part 3)
- generating source code based upon the recovered public and private object state data (XMLBean can transform an XML document...into a running JavaBean,” pag 1 par 2 in Part 3)
- using the generated source code to perform a computer operation (XMLBean can transform an XML document...into a running JavaBean,” pag 1 par 2 in Part 3)

Per claim 30:

The rejection of claim 26 is incorporated, and further, Johnson teaches that the public and private object state data comprise state data from JavaBeans (“an XML file that specifies the values for a JavaBean’s properties,” pg 7 in Part 1).

Per claim 31:

The rejection of claim 26 is incorporated, and further, Johnson teaches:

- parsing the XML tags to recover the public and private object state data  
(“readXMLBean ... creates an XML parser, reads the input the input XML file,” pg 12 par 3 in Part 3)
- instantiating objects in an order based upon the stored state restoration order  
(“allowing specification of a setAsDOM() method to match any getAsDOM()”

Art Unit: 2124

method...XMLBeanReader uses setAsDOM() to initialize a property value in a bean-object instance from the structure of a DOM subtree," pg 5 par 1-3 in Part 3; pg 9 par 3-7 in Part 1), wherein the instantiating of the object recovers the public and private object state data ("identifies the first element of the document read, and calls instantiateBean() to create an instance of the bean and initialize it. This method parses XML into a DOM tree by calling a canned parser," pg 12 par 3-4 in Part 3)

-using the instantiated objects within the object development environment (see Listing 4. readXMLBean() parses XML and calls instantiateBean() in Part 3).

Per claim 1:

Johnson discloses:

- determining the private object state data of objects used within the object development environment("XML file that specifies the values for a JavaBeans' properties," pg 7 paragraphs 2-8; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- storing the determined private object state data in a computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2)
- wherein the computer-readable file is in a human-understandable format ("making JavaBeans mobile and interoperable by representing them as XML documents," pg 1 summary;
- restoring the private object state data by processing the computer-readable file ("converts XML files to JavaBeans running in memory," pg 1 summary)



Per claim 2:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the human-understandable format is a text-based format ("Beans as XML documents," pg 7 paragraph 2)

Per claim 3:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- after storing the determined private object state data in the computer-readable file, modifying the private object state data within the computer-readable file without using the object development environment ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)
- restoring the stored private object state data by processing the computer-readable file, wherein the restored private object state data contains the modifications to the private object state data ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; "We simply gave the XML parser the name of the XML file, and it returned the entrie data structure that the XML file represented," pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)

Per claim 4:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

Art Unit: 2124

- after storing the determined private object state data in the computer-readable file, directly editing the computer-readable file in order to modify the private object state data within the computer-readable file("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)
- -restoring the stored private object state data by processing the computer-readable file, wherein the restored private object state data contains the modifications to the private object state data ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

Per claim 5:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the modifications are to correct errors in object structure without using the object development environment (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section).

Per claim 6:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- an object class that specifies structure of the private object state data has been modified(pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)
- said modification occurring after the private object state data has been stored in the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section).

Art Unit: 2124

-restoring the private object state data from the computer-readable file even though the object class has been modified (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 7:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- an object class that is included in the object development environment and that specifies structure of the private object state data has been modified (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)
- said modification occurring after the private object state data has been stored in the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)
- restoring back into the object development environment the private object state data from the computer-readable file despite the structures differing between the modified class and the private object state data (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 8:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- determining public and the private object state data of the objects used within the object development environment ("XML file that specifies the values for a JavaBeans' properties," pg 7 paragraphs 2-8)

Art Unit: 2124

- storing the determined public and private object state data in the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2)
- restoring the private and public object state data by processing the computer-readable file ("converts XML files to JavaBeans running in memory," pg 1 summary)

Per claim 9:

The rejection of claim 1 is incorporated, and further, Johnson teaches a Java development environment ("JavaBeans," pg 7 paragraph 2 in Part 1).

Per claim 10:

The rejection of claim 1 is incorporated, and further, Johnson teaches a Java development environment for providing graphical user interfaces ("JavaBeans," pg 7 paragraph 2 in Part 1)

Per claim 11:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the computer-readable file is in a structured format that indicates order in which the private object state data is to be restored ("using XML as a persistence format for Javabeans components" pg 2 paragraph 5-6 in Part2)

Per claim 12:

The rejection of claim 11 is incorporated, and further, Johnson teaches:

Art Unit: 2124

restoring the private object state data in the order specified by the structured format of the computer-readable file (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 13:

The rejection of claim 12 is incorporated, and further, Johnson teaches that the structured format is an XML structured format (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 14:

The rejection of claim 13 is incorporated, and further, Johnson teaches that the XML structured format includes nested XML blocks to indicate the order in which the private object state data is to be restored (See Figure 8 in Part 2).

Per claim 15:

The rejection of claim 1 is incorporated, and further, Johnson teaches that the computer-readable file is in a structured format that contains the private and public object state data ("using XML as a persistence format for JavaBeans components" pg 2 paragraph 5-6 in Part2).

Per claim 16:

The rejection of claim 1 is incorporated, and further, Johnson teaches restoring into a different type of object development environment the private object state data from the computer-readable file (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Art Unit: 2124

Per claim 17:

The rejection of claim 16 is incorporated, and further, Johnson teaches an XML structure such that the computer-readable file is capable of being imported both into the object development environment and the different type of object development environment (pg 2 paragraph 5-6 in Part2; pg 4 "Customization" in Part 3).

Per claim 18:

The rejection of claim 16 is incorporated, and further, Johnson teaches:  
creating Java objects based upon the restored private object state data, wherein the Java objects are used within the object development environment (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 19:

The rejection of claim 16 is incorporated, and further, Johnson teaches:

- creating Java objects based upon the restored private object state data (see the section "Creating JavaBeans from XML," pg 9 in Part 1)
- storing the Java objects in an object hash table ("The object being created is introspected to get a list of its properties, and then a hash table is created that indexes property names against their corresponding property descriptors," pg 14 paragraph 5 in Part 3)
- retrieving a frame based upon the Java objects stored in the object hash table (pg 14 paragraph 5 in Part 3 )

Per claim 20:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- restoring the private object state data by processing the computer-readable file (see the section "Creating JavaBeans from XML," pg 9 in Part 1)
- using the restored private object state data to generate source code (see the section "Creating JavaBeans from XML," pg 9 in Part 1)

Per claim 21:

The rejection of claim 20 is incorporated, and further, Johnson teaches:

using the restored private object state data to generate a different type of source code (pg 2 paragraph 6-7 in Part 3; pg 4-5, Customization section)

Per claim 22:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

- (a) determining whether the private object state data of the objects have been modified from the initial values given to the objects upon the objects' creation (pg 4-5, Customization section; pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)
- (b) storing in the computer-readable file the private object state data that has been determined in step (a) to have been modified ("hook functions" that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section in Part 3; pg 2 paragraph 6-7 in Part 3)

-(c) restoring from the computer-readable file the private object state data that has been determined in step (a) to have been modified (pg 4-5, Customization section; pg 3 paragraph 3 in Part 2; pg 2 paragraph 6-7 in Part 3)

Per claim 23:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-determining customization hooks associated with the objects used within the object development environment ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

-storing the customization hooks in the computer-readable file ("hook functions that allow a programmer to customize the package's behavior without doing violence to its operation," pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

-restoring the customization hooks by processing the computer-readable file (pg 4-5, Customization section; pg 2 paragraph 6-7 in Part 3)

Per claim 24:

The rejection of claim 1 is incorporated, and further, Johnson teaches:

-determining design time object state data associated with the objects used within the object development environment ("use XML as a serialization format for beans," pg 7 paragraph 2)



Art Unit: 2124

- storing the design time object state data in the computer-readable file("use XML as a serialization format for beans," pg 7 paragraph 2)
- restoring the design time object state data by processing the computer-readable file, wherein the restored design time object state data is used during design time ("converts XML files to JavaBeans running in memory," pg 1 summary)

Per claim 25:

The rejection of claim 24 is incorporated, and further, Johnson teaches:

- determining run time object state data associated with the objects used within the object development environment ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- storing the run time object state data in the computer-readable file ("use XML as a serialization format for beans," pg 7 paragraph 2; pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3)
- restoring the run time object state data by processing the computer-readable file, wherein the restored run time object state data is used during run time (pg 2 paragraph 6-7 in Part 3; pg 9 last paragraph –first paragraph in pg 10 in Part 3).

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 703-305-6465. The examiner can normally be reached on M-F 8:30-5:30.

Art Unit: 2124

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 703-305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

IK  
3/2/2004

*Kakali Chaki*  
**KAKALI CHAKI**  
**SUPERVISORY PATENT EXAMINER**  
**TECHNOLOGY CENTER 2100**